Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

**Лабораторная работа №2**

**«Модульное тестирование библиотеки классов**

**на C# средствами Visual Studio»**

**Вариант №4**

Выполнил: студент 4 курса

ИВТ, гр. ИП-113

Шпилев Д. И.

Проверил: старший преподаватель кафедры ПМиК

Агалаков А.А.

Новосибирск, 2024 г.

**Цель**

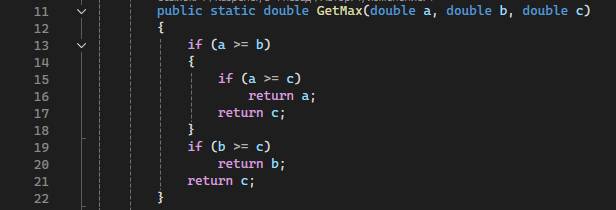
Сформировать практические навыки разработки модульных тестов для библиотек классов C# и выполнения модульного тестирования с помощью средств автоматизации Visual Studio.

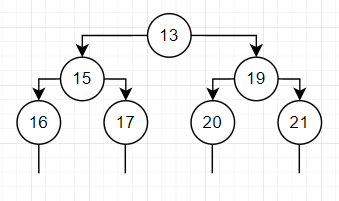
**Задание**

Разработайте на языке С# класс, содержащий функции в соответствии с вариантом задания. Разработайте тестовые наборы данных для тестирования функций класса, по критерию С1. Протестируйте созданный класс с помощью средств автоматизации модульного тестирования Visual Studio. Проанализируйте результаты выполненных тестов по объёму покрытия тестируемого кода. Напишите отчёт о результатах проделанной работы.

1. Поиск максимума из трёх чисел.
2. Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает произведение значений компонентов массива, у которых сумма значений индексов – чётная.
3. Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает минимальное значение компонентов массива, лежащих на и ниже главной диагонали

**УГП и тестовые наборы данных для тестирования функций класса**

****

****

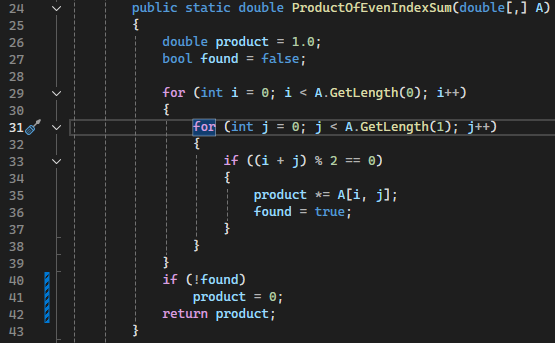
Пути:

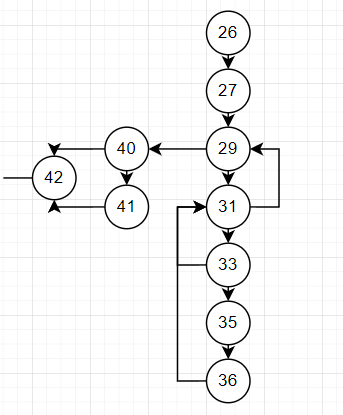
(13-15-16) (a = 133.5; b = 133.5; c = 133.5;)

(13-15-17) (a = 133.5; b = 51.2; c = 183.3;)

(13-19-20) (a = 3.5; b = 51.2; c = 12.3;)

(13-19-21) (a = 3.5; b = 5; c = 12.3;)

****

****

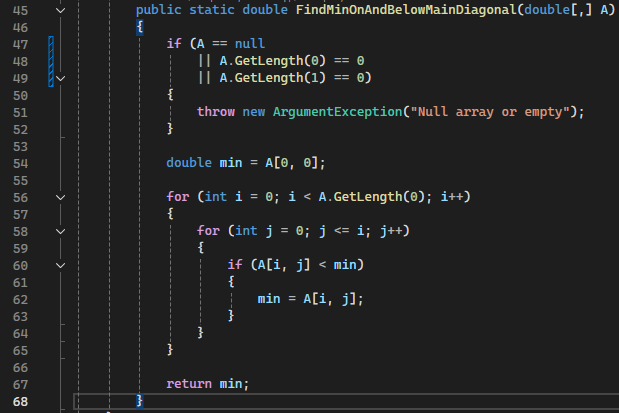
Пути (26-27-29-40-41-42) (26-27-29-40-42) (26-27-29-31-40-41-42) (26-27-29-31-40-42) (26-27-29-31-33-40-41-42) (26-27-29-31-33-40-42) (26-27-29-31-33-35-36-40-41-42) (26-27-29-31-33-35-36-40-42)

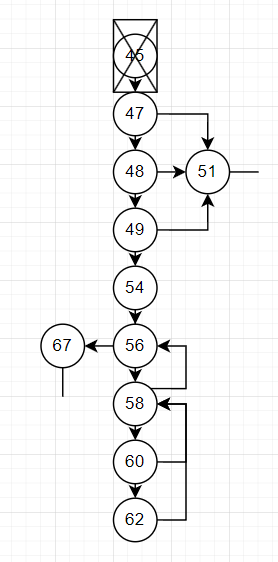
Тесты:

A = { { 5.0 } };

A = {{ 2.0, 3.0 }, { 4.0, 5.0 }};

A = { { } };





Для полного тестирования нужно выполнить исключения из пунктов 47, 48, 49 и попасть в пункт 51, пройти путь 56-58-60-62-67.

Для исключения берем A = null; пустой массив A = { }; и A = { { } };

A = {{ 5.0, 3.2 }, { 3.5, 5.2 }};

A = { { 5.0 } };

**Листинг программы:**

**Program.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

public class Program

{

public static double GetMax(double a, double b, double c)

{

if (a >= b)

{

if (a >= c)

return a;

return c;

}

if (b >= c)

return b;

return c;

}

public static double ProductOfEvenIndexSum(double[,] A)

{

double product = 1.0;

bool found = false;

for (int i = 0; i < A.GetLength(0); i++)

{

for (int j = 0; j < A.GetLength(1); j++)

{

if ((i + j) % 2 == 0)

{

product \*= A[i, j];

found = true;

}

}

}

if (!found)

product = 0;

return product;

}

public static double FindMinOnAndBelowMainDiagonal(double[,] A)

{

if (A == null

|| A.GetLength(0) == 0

|| A.GetLength(1) == 0)

{

throw new ArgumentException("Null array or empty");

}

double min = A[0, 0];

for (int i = 0; i < A.GetLength(0); i++)

{

for (int j = 0; j <= i; j++)

{

if (A[i, j] < min)

{

min = A[i, j];

}

}

}

return min;

}

}

}

**UnitTest1.cs:**

using Lab2;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

namespace UnitTestLab2

{

[TestClass]

public class GetMax

{

[TestMethod]

public void GetMaxC()

{

//arrange

double a = 3.5;

double b = 5;

double c = 12.3;

//act

double result = Lab2.Program.GetMax(a, b, c);

//assert

Assert.AreEqual(c, result);

}

[TestMethod]

public void GetMaxB()

{

//arrange

double a = 3.5;

double b = 51.2;

double c = 12.3;

//act

double result = Lab2.Program.GetMax(a, b, c);

//assert

Assert.AreEqual(b, result);

}

[TestMethod]

public void GetMaxA()

{

//arrange

double a = 133.5;

double b = 133.5;

double c = 133.5;

//act

double result = Lab2.Program.GetMax(a, b, c);

//assert

Assert.AreEqual(a, result);

}

[TestMethod]

public void GetMaxC2()

{

//arrange

double a = 133.5;

double b = 51.2;

double c = 183.3;

//act

double result = Lab2.Program.GetMax(a, b, c);

//assert

Assert.AreEqual(c, result);

}

}

[TestClass]

public class ProductOfEvenIndexSum

{

[TestMethod]

public void OneElement()

{

//arrange

double[,] A = { { 5.0 } };

//act

double result = Lab2.Program.ProductOfEvenIndexSum(A);

//assert

Assert.AreEqual(5.0, result);

}

[TestMethod]

public void TwoSizeLenghts()

{

//arrange

double[,] A = {

{ 2.0, 3.0 },

{ 4.0, 5.0 }

};

//act

double result = Lab2.Program.ProductOfEvenIndexSum(A);

//assert

Assert.AreEqual(10.0, result);

}

[TestMethod]

public void ZeroElements()

{

//arrange

double[,] A = { { } };

//act

double result = Lab2.Program.ProductOfEvenIndexSum(A);

//assert

Assert.AreEqual(0, result);

}

}

[TestClass]

public class FindMinOnAndBelowMainDiagonal

{

[TestMethod]

public void OneElement()

{

//arrange

double[,] A = { { 5.0 } };

//act

double result = Lab2.Program.FindMinOnAndBelowMainDiagonal(A);

//assert

Assert.AreEqual(5.0, result);

}

[TestMethod]

public void ManyElement()

{

//arrange

double[,] A = {

{ 5.0, 3.2 },

{ 3.5, 5.2 }

};

//act

double result = Lab2.Program.FindMinOnAndBelowMainDiagonal(A);

//assert

Assert.AreEqual(3.5, result);

}

[TestMethod]

public void NullArray()

{

//arrange

//act

//assert

Assert.ThrowsException<ArgumentException>(() => Lab2.Program.FindMinOnAndBelowMainDiagonal(null));

}

[TestMethod]

public void ZeroLengthArray()

{

//arrange

double[,] A = { };

//act

//assert

Assert.ThrowsException<ArgumentException>(() => Lab2.Program.FindMinOnAndBelowMainDiagonal(A));

}

[TestMethod]

public void ZeroLengthSubArray()

{

//arrange

double[,] A = { { } };

//act

//assert

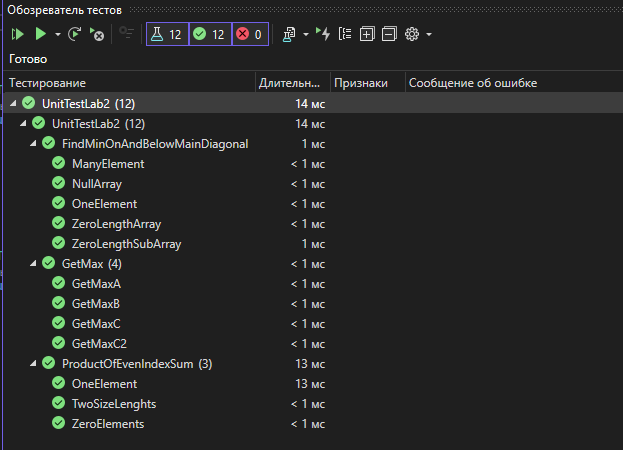
Assert.ThrowsException<ArgumentException>(() => Lab2.Program.FindMinOnAndBelowMainDiagonal(A));

}

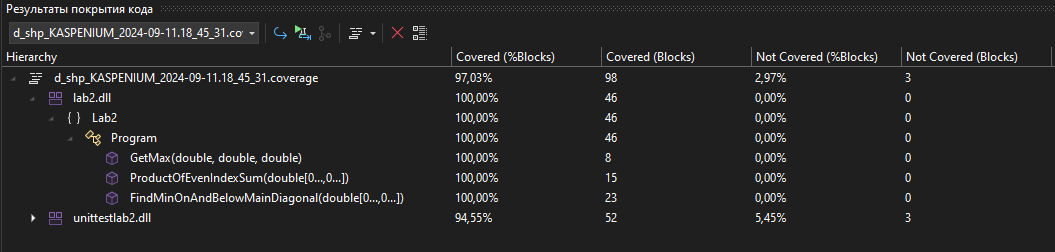
}

}

**Результаты выполнения модульных тестов**

****

**Результаты покрытия разработанного кода тестами.**



**Вывод**

Были сформированы практические навыки разработки модульных

тестов для библиотек классов C# и выполнения модульного тестирования с

помощью средств автоматизации Visual Studio, разработан класс на языке C#,

содержащий функции в соответствии с вариантом задания, разработаны

тестовые наборы данных для тестирования функций класса, по критерию С1,

не были получены результаты покрытия кода тестами в связи с ограничением

версии Visual Studio.